CENTROID-BASED JOINT REPRESENTATION FOR POSE ESTIMATION AND INSTANCE SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Joint pose estimation and instance segmentation combines keypoint heatmaps with segmentation masks for multi-person pose and instance-level segmentation. Unlike easy cases with explicit heatmap activation, hard cases with implicit heatmap due to multi-person entanglement, overlap, and occlusions requires joint representation with a segmentation mask in end-to-end training. This paper presents a new centroid-based joint representation method called CENTERFOCUS. It follows a bottom-up paradigm to generate Strong Keypoint Feature Maps for both soft and hard keypoints and improve keypoints detection accuracy as well as the confidence score by introducing KeyCentroids and a Body Heat Map. CENTERFOCUS then uses the high-resolution representation of keypoint as a center of attraction for the pixels in the embedding space to generate MaskCentroid to cluster the pixels to a particular human instance to whom it belongs, even if 70% of the body is occluded. Finally, we propose a new PoseSeg algorithm that collects the feature representation of a 2D human pose and segmentation for the joint structure of the pose and instance segmentation. We then experimentally demonstrate the effectiveness and generalization ability of our system on challenging scenarios such as occlusions, entangled limbs, and overlapping people. The experimental results show the effectiveness of CENTERFOCUS outperforms representative models on the challenging MS COCO and OCHuman benchmarks in terms of both accuracy and runtime performance, Ablation experiments analyze the impact of each component of the system. The code will be released publicly.

1 INTRODUCTION

Joint pose estimation and body segmentation are widely used for human-computer interactions and real-time image/video analytics. The main goal is to identify individuals and their activities from 2D positioning of human joints and their body shape structure. There are two primary challenges in multi-person joint pose estimation and instance segmentation: (i) an unknown number of individuals are overlapped, occluded, or have entangled limbs, and (ii) computational complexity increases with the number of individuals. An image can have an undefined number of individuals at any location and distance. Moreover, human-to-human interactions, especially for those who are socially engaged, incur complex spatial interference because of contacts, obstructions, and articulation of their limbs, making it difficult to associate body parts. As a result, the computational cost and complexity increase rapidly with the number of people in the image, necessitating an efficient, scalable, and accurate pose and segmentation model.

Existing proposals for pose estimation He et al. (2017); Chen et al. (2018); Fang et al. (2017); Huang et al. (2017); Li et al. (2019) acknowledge these challenges and rely on a top-down approach to first detect people in the image and then estimate the pose of each detected person. Recent studies by He et al. (2017); Papandreou et al. (2018) suggest that large-scale pose and segmentation datasets (e.g., COCO, Lin et al. (2014), OCHuman Zhang et al. (2019)) enable joint estimation of the human pose and body segmentation producing state-of-the-art (SOTA) results for both tasks. However, these top-down approaches Chen et al. (2018); Fang et al. (2017); Huang et al. (2017); Li et al. (2019) require a pose estimator to run iteratively for each detected person, resulting in severe runtime performance. Moreover, establishing a segmentation head using a top-down approach increases the computational cost He et al. (2017), rendering them infeasible for real-time applications.

This paper presents CENTERFOCUS, a new centroid-based joint representation for human pose estimation and instance segmentation model. CENTERFOCUS uses a bottom-up approach to first detect

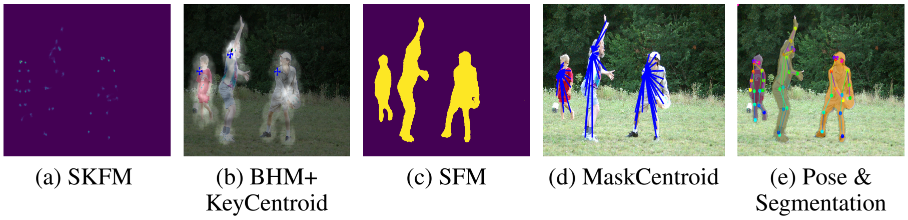

(a) SKFM (b) BHM+ KeyCentroid (c) SFM (d) MaskCentroid (e) Pose & Segmentation

Figure 1: CENTERFOCUS produces (a) a strong keypoint feature map (SKFM) to detect each individual keypoint, (b) a body heat map (BHM) along with the KeyCentroid to improve keypoint accuracy in crowds, (c) semantic feature map (SFM) for each individual, (d) MaskCentroid to improve pixel-level classification and instance-level segmentation, and (e) Finally pose & instance segmentation.

keypoints employing the pose head network for pose estimation and then perform pixel-level classification employing the segmentation head network where the detected keypoints are used as a center of attraction to associate the pixels to the right instance. Unlike top-down approaches Chen et al. (2018); Fang et al. (2017); Huang et al. (2017); Li et al. (2019), CENTERFOCUS detects the human body without requiring a box detector or incurring runtime complexity.

CENTERFOCUS is not the first method to jointly perform human pose estimation and instance segmentation Ahmad et al. (2022); He et al. (2017); Zhang et al. (2019) or leverage bottom-up approaches Papandreou et al. (2018). However, existing models Papandreou et al. (2018) use human poses to refine pixel-wise clustering for segmentation and thus do not perform segmentation task well. Moreover, they have high overheads because of the extra computation of a person detector He et al. (2017), scalability issues for instance segmentation Zhang et al. (2019), and a high computational cost Ahmad et al. (2022), which makes them unsuitable for crowd scenarios and real-time applications. Unlike existing models, CENTERFOCUS does not incur the high overheads associated with top-down approaches because of the person detector, or the segmentation performance and scalability concerns associated with bottom-up approaches due to pixel-wise clustering. Instead, CENTERFOCUS leverages high confident keypoints for centroid-based joint representation for both pose and segmentation tasks.

CENTERFOCUS addresses the aforementioned challenges using two primary networks: the pose head network and the segmentation head network. The pose head network generates an SKFM that estimates the relative displacement between pairs of keypoints and improves the precision of the long-range, occluded, and proximate keypoints (Figure 1a). Using extracted keypoint features, a KeyCentroid is produced to define 2D offset vectors for each pixel that points at the center of attraction for each keypoint, helping CENTERFOCUS to identify the precise human keypoint coordinates. Along with the KeyCentroid, CENTERFOCUS generates BHM using the SKFM, helps CENTERFOCUS to increase the intensity of each keypoint and improve the keypoint detection confidence score (Figure 1b). The segmentation head network performs pixel-level classification and generates SFM for each individual (Figure 1c) using the MaskCentroid. The MaskCentroid defines the embedding space to associate pixels to the right instance by defining the keypoint as a centroid (Figure 1d). The MaskCentroid helps to produce an instance-level classification for the human class. At last, we designed a new PoseSeg algorithm that utilizes all the components of the system to present the human pose and instance-level segmentation (Figure 1e).

We evaluated the performance of CENTERFOCUS using the COCO Lin et al. (2014) and OCHuman Zhang et al. (2019) datasets. To the best of our knowledge, CENTERFOCUS is the first reliable application for the task of human pose estimation and instance segmentation. This paper makes the following contributions.

- KeyCentroid defines 2D offset vectors points to the center coordinates in the keypoint feature map, that helps to identify the precise keypoint coordinates for pose estimation (§3.2);
- MaskCentroid defines the keypoint as a centroid for the 2D offset vectors in the embedding space, that helps to associate the pixels to the right instance to perform pixel-level instance segmentation (§3.3);
- Our in-depth evaluation (§4) and ablation experiments (§5) demonstrate the effectiveness in human pose and instance segmentation.

2 Related Work

Human Pose Estimation. Human pose estimation uses two main techniques: top-down and bottom-up methods. The top-down approach identifies keypoints surrounded by a bounding box detector. Representative works include: HRNet Cheng et al. (2020), RMPE Fang et al. (2017), Multiposenet Kocabas et al. (2018), Hourglass Newell et al. (2016), convolutional pose machine Wei et al. (2016), CPN Chen et al. (2018), Mask r-cnn He et al. (2017), simple baseline Xiao et al. (2018), CSM-SCARB Su et al. (2019), RSN Cai et al. (2020), and Graph-PCNN Wang et al. (2020). These methods explore the human pose in a person detector, thus achieving satisfactory performance; however, the person box detection is costly. The bottom-up methods detect the keypoint in one shot some pioneering methods, such as DeepCut Pishchulin et al. (2016) and DeeperCut Insafutdinov et al. (2016). These methods formulate the association between keypoints as an integer linear scheme and require a longer processing time. Other part-affinity field techniques like OpenPose Cao et al. (2017) and other extensions, such as Pif-Paf Kreiss et al. (2019), associative embedding Newell et al. (2017), PersonLab Papandreou et al. (2018), and HGG Jin et al. (2020) are developed based on grouping techniques that often fail in crowded scenarios. We aim to improve hard keypoint detection performance by introducing the SKFM, KeyCentroid, and BHM.

Instance Segmentation. There are two primary approaches for instance-level segmentation: (1) single-stage instance segmentation Dai et al. (2016); Long et al. (2015); Bolya et al. (2019) and (2) multi-stage instance segmentation He et al. (2017); Ren et al. (2015). The single-stage approach creates intermediate and distributed feature maps based on the entire image. The InstanceFCN Dai et al. (2016) create several instance-sensitive scoring maps and apply the assembly module to the output instance. This approach is faster than the multi-stage approach; however, it requires repooling and other non-trivial computations (e.g., mask voting) that affect real-time processing. YOLACT Bolya et al. (2019) generate a set of prototype masks and then use coefficients per mask to produce the instance-level segmentation; however, it is critical to obtain a high resolution. Multi-stage instance segmentation follows the detect-then-segment paradigm. This approach first performs bounding box detection, and then the pixels are classified to obtain the final mask in the bounding box region. Mask R-CNN He et al. (2017) is based on multi-stage instance segmentation that extends Faster R-CNN Ren et al. (2015) by adding a branch for predicting segmentation masks for each Region of Interest (RoI). The method presented by Liu et al. (2018) improves the accuracy of the Mask R-CNN by enriching the Feature Pyramid Network (FPN) features.

Pose Estimation and Instance Segmentation. SOTA developments have been made in human pose estimation and instance segmentation. Mask R-CNN He et al. (2017) was the first pioneer method; however, suffers from heavy computational cost. Pose2Seg Zhang et al. (2019) proposed human pose-based instance segmentation. This method separates instances based on the human pose, rather than the proposal region. It takes already generated pose as input that makes concerns an end-to-end training model. PersonLab Papandreou et al. (2018) group keypoints by using greedy decoding. This method also reports a part-induced geometric embedding descriptor for human class instance segmentation. However, this approach fails to perform segmentation on highly entangled instances. A new Pose*Plus*Seg Ahmad et al. (2022) played an important role in this regard; however, it compromises a couple of backbone and refined networks making it a complex structure model. We propose a simple, yet effective system to handle the above complications by introducing MaskCentroid that defines a keypoint as a centroid in the embedding space to associate the pixels to the right instance and improve the segmentation performance.

3 Methods

CenterFocus comprises one core pipeline, as shown in Figure 2. First, a ResNet backbone is used to extract the features from the image. The learned feature maps are then fed into the pose and segmentation head. The pose head is designed based on the same residual unit in ResNet He et al. (2016) allowing CenterFocus to generate the SKFM from the learned features and present it as a human skeletal structure (§3.1). KeyCentroid and BHM are introduced to improve the keypoint prediction and confidence score (§3.2). The segmentation head is designed as the same residual unit in ResNet He et al. (2016) and performs pixel-level classification with mask features for each human instance. A MaskCentroid is defined to better align the 2D shape by defining the keypoint as a center of attraction to associate the pixels to the right instance in the embedding space to predict instance-level segmentation (§3.3). CenterFocus uses the features from both heads as input to the PoseSeg module and generates the final human pose and instance segmentation.

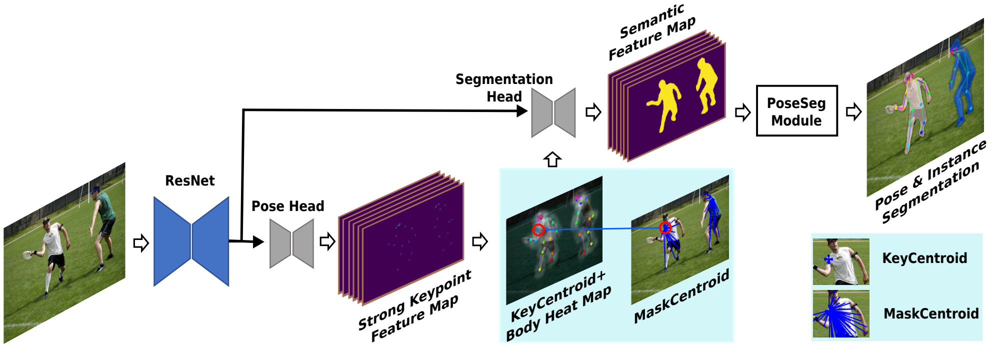


Figure 2: CENTERFOCUS comprising one core pipeline where the pose head generates a strong keypoint feature map (SKFM) and KeyCentroid to predict the optimal 2D keypoint coordinates localized by a body heat map (BHM) to enrich the keypoint confidence score. The segmentation head is designed to produce a semantic feature map (SFM) using the MaskCentroid as a center of attraction for the pixels in the embedding space to assign the pixel to the right instance. Finally, the PoseSeg Module uses the information of both heads for the final output, i.e., human pose and instance segmentation.

3.1 STRONG KEYPOINT FEATURE MAP

CENTERFOCUS generates SKFM by employing the pose head network, as illustrated in Figure 2, as the base for the pose estimation. In this stage, each individual keypoint is detected and concatenated for the output feature maps. Specifically, we adopt the residual-based network for our multi-person pose setting to produce SKFM, one channel per keypoint and KeyCentroid two channels per keypoint for the vertical and horizontal displacement.

Let p_i represent the keypoint position in the image, where $i = \{1, \ldots, N\}$ are mapped to the 2D positions of the pixels. A keypoint disk $D_R(q) = \{p : \|p–q\| \leq R\}$ of radius R is focused at point q. Additionally, let $q_{j,k}$ be the 2D position of the j^{th} keypoint of the k^{th} person instance, where $j = \{1, \ldots, I\}$ and I is the number of individual keypoints in the image. For each known keypoint j, a binary classification approach is followed. Specifically, every predicted keypoint pixel p_i is binary classified such that p_i=1 if $p_i \in D_R$ for each person keypoint j, otherwise p_i=0. Thus, for every keypoint, there are independent dense binary classification tasks. To obtain the SKFM for each keypoint j, we define a disk D_R of radius $R = 32$ (diameter = 64) independent of the keypoint scale. To equally weigh person keypoints in the classification loss, we choose a disk radius that does not scale according to the instance size. Note that R is constant for all experiments in this paper for optimal results. While training the network, SKFM loss is computed based on the annotated image positions, back-propagating across the entire image, excluding the regions with individuals who are not fully annotated with keypoints (e.g., crowded areas and small individual segments).

Point-wise Gaussian Optimization. To achieve impact-full coordinates of keypoints we employ Gaussian smooth Chung (2020) for each individual keypoint, i.e., *point-wise Gaussian optimization*, to reduce the noise and retain the useful information while producing the SKFM. such as:

$$Gq(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{1}$$

where σ is the standard deviation of the distribution and x,y represent the 2D keypoint coordinates. To handle the variation between keypoints, we set the σ range from 0 to 1. For High Variant Keypoints (HVK) (e.g., wrist, ankle, elbow, and knee), we set $0.1 \leq \sigma < 0.5$; however, for Low Variant Keypoints (LVK) (e.g., nose, shoulder, hip), we set $0.5 \leq \sigma < 1$, as shown in Figure 3. A σ near to 0 increases the pixel intensity of the keypoints and works better in crowded and entangled scenarios. A σ near to 1 works best in non-crowded cases. We analyze how the values of σ contribute to the performance of the system in ablation (§5.4).

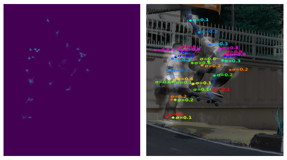


Figure 3: Point-wise Gaussian optimization where σ values are defined for each keypoint.

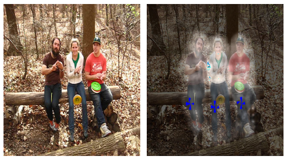


Figure 4: KeyCentroid defined for right knees. BHM is generated using the keypoint disk.

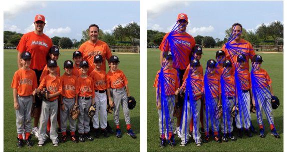


Figure 5: MaskCentroid of each individual. The right shoulder is the center of attraction.

3.2 KeyCentroid and Body Heat Map

In addition to SKFM, our pose head along with the residual network defines KeyCentroid k_c per keypoint. The purpose of KeyCentroid is to improve keypoint localization accuracy. For each keypoint pixel p_i within the keypoint disk D_R, the 2D KeyCentroid vector $k_v = q_{j,k} - p_i$ locus from the position x in the image to the j^{th} keypoint of the k person instance, as illustrated in Figure 4. We generate a number of vector fields in the keypoint disk D_R by solving a 2D regression problem at each keypoint position p_i. During training, we penalize the k_c error by $L1$ loss and back-propagate the error at position $p \in D_R$ in the keypoint disk. The disk radius is fixed as $R = 32$ to normalize the KeyCentroid and make its dynamic range equivalent to the SKFM loss. We aggregate the SKFM and KeyCentroid for the optimal keypoint coordinates generally for soft and specifically for hard keypoints. Our ablation experiments analyze the impact of SKFM and KeyCentroid on keypoint detection (§5.1).

Body Heat Map. Along with KeyCentroid, we use D_R to produce a BHM as shown in Figure 4 that captures the important parts of the body (e.g., leg, torso, head, and hand), working like a fine-grained human detector rather than a box detector Zhou & Yuan (2017; 2018). BHM specifically enriches the keypoint prediction confidence score and helps CenterFocus to select a visible keypoint for the MaskCentroid. BHM is produced utilizing the keypoint disk D_R, where radius R=32 for all predicted pixels p_i belonging to the D_R for every individual k. p_i is a group of pixels that represents the keypoint disk. The BHM representation is summarized in Eq. 2.

$$k = \sum_{i=1}^{n} p_i(\alpha), \text{where } p_i = 2\pi R. \tag{2}$$

α ranges from 0 to 1 to maintain the bright resolution of each keypoint disk. During inference, the BHM helps to detect individuals and also increases the keypoint confidence score. In our experiments, we analyze the impact of BHM on the keypoint confidence score (§5.2).

3.3 MaskCentroid

Human instance segmentation is a pixel-level classification challenge where to connect pixels with the right person instance I. For this task, we define MaskCentroid as illustrated in Figure 5 to cluster the mask pixels with the defined centroid C_i inside each annotated person instance with 2-D mask pixels, which points from image position x_i to the position of C_i of the corresponding instance. At each image position x_i of a semantically identified human instance, the embedding vector $e(x_i)$ reflects a local approximation of the absolute location of each mask pixel of an individual to whom it corresponds, i.e., it represents the person's expected shape. To this end, for each pixel, we learn the pixel offset, which points to C_i (right shoulder). Here, we take advantage of the high confident keypoint localization to use them as a center of attraction for each instance pixel. The purpose of instance segmentation is to cluster a set of pixels $P = \{m_0, m_1, m_2, ..., m_i\}$ and its 2-D embedding vectors $e(m_i)$, into a set of instances $I = \{n_0, n_1, n_2, ..., n_j\}$ to provide a 2D mask for the human instance. Pixels are assigned to their corresponding centroid:

$$C_i = \frac{1}{N} \sum_{m_i \in n_j} m_i. \tag{3}$$

This is attained by defining pixel offset vector v_i for each known pixel m_i, so that the resulting embedding $e_i = m_i + v_i$ points from its respective instance centroid. We penalize pixel offset loss by the L_1 loss function throughout model training, averaging and back-propagating at the image position x_i that corresponds to an instance of a specific individual entity:

$$L = \sum_{i=0}^{n} \|v_i - \hat{v}_i\|, \tag{4}$$

where $\hat{v}_i = C_i - n_j$ for $n_j \in m_i$. In order to cluster the pixels to their centroid, it is important to specify the positions of the instance centroids and to assign pixels to a particular instance centroid. We use a density-based clustering algorithm Kriegel et al. (2011) to first locate a set of centroids as a center of attraction. Having obtained an array of centroids $\mathcal{C} = \{C_0, C_1, ..., C_K\}$, we add pixels to a particular instance based on a minimum distance-to-centroid:

$$e_i \in m_i : k = \text{ arg min}_{\mathcal{C}} \|e_i - C\|. \tag{5}$$

During inference, the MaskCentroid effectively addresses the challenging scenarios where 70% of the human body is occluded. Our experimental study demonstrates the effectiveness of the MaskCentroid on human instance segmentation (§5.3).

Instance-wise Gaussian Optimization. Once the segmentation head performs pixel-level classification and identifies the individual semantically, CENTERFOCUS performs Gaussian smoothing Chung (2020) at the instance level, i.e., *instance-wise Gaussian optimization*. We apply instance-wise smoothing individually using the σ ranging from 0 to 1. Based on our experiments, σ near 0.1 produces a more precise segmentation mask when individuals are entangled and overlapping. The ablation experiment demonstrates the observation of instance-wise smoothing (§5.4).

4 EVALUATION

We evaluate CENTERFOCUS using the standard benchmarks, COCO Lin et al. (2014) and OCHuman Zhang et al. (2019), focus on heavily occluded individuals and compare the computational cost and inference time with SOTA models. The model is trained end-to-end using the COCOPersons training set. Ablations are conducted on the COCO *val* set. We used the CNN backbone networks ResNet-101 and ResNet-152 He et al. (2016) for training and testing. The hyperparameters for training were: learning rate = $0.1 \times e^{-4}$, image size $= 401 \times 401$, batch size $= 4$, and Adam optimizer. We performed various transformations during model training, such as scale, flip, and rotate operations. We conducted synchronous training for 400 epochs with stochastic gradient descent using TensorFlow 1.13 on a single TITAN RTX.

Models	Backbone	AP	AP$^{.50}$	AP $^{.75}$	AP^{M}	AP^{L}	AR
Top-down:							
8-stage Hourglass	-	0.671	-	-	-	-	-
CPN	ResNet-50	0.727	-	-	-	-	-
SimpleBaseline	ResNet-152	0.720	0.893	0.798	0.687	0.789	0.778
Bottom-up:							
CMU-Pose $*$	-	0.610	0.849	0.675	0.563	0.693	-
PersonLab	ResNet-152	0.665	0.862	0.719	0.623	0.732	0.707
PifPaf ×	-	0.674	-	-	-	-	-
HGG †	Hourglass	0.683	0.867	0.758	-	-	0.720
Point-Set Anchors †	HRNet-W48	0.698	0.888	0.763	0.659	0.766	0.756
DEKR48	HRNet-W48	0.723	0.883	0.786	0.686	0.786	0.777
Pose*Plus*Seg	ResNet-152	0.744	0.894	0.748	0.675	0.811	0.810
CENTERFOCUS (ours)	ResNet-101	0.735	0.898	0.819	0.715	0.797	0.798
CENTERFOCUS (ours)	ResNet-152	0.749	0.899	0.824	0.718	0.816	0.815

Table 1: Performance comparison using the **COCO** keypoint *val* set. $*$ indicates refinement. × indicates single-scale testing † indicates multi-scale testing. AP is at IOU=0.5:0.05:0.95, AP$^{0.50}$ at IOU=0.50 (Pascal VOC metric), AP$^{0.75}$ at IOU=0.75 (strict metric), AP$^{\mathrm{M}}$ corresponds to AP for medium objects: 32^2 <area <96^2, and AP$^{\mathrm{L}}$ corresponds to AP for large objects: area >96^2.

Keypoint Detection Results. Table 1 summarizes the results on the COCO *val* dataset. CENTERFOCUS (0.749 mAP) outperformed top-down 8-stage Hourglass Newell et al. (2016), CPN Chen et al. (2018), and SimpleBaseline Xiao et al. (2018). CENTERFOCUS also surpassed bottom-up models: pifpaf Kreiss et al. (2019) by 0.075 AP, HGG Jin et al. (2020) by 0.066 AP, Point-Set Anchors Wei et al. (2020) by 0.051 AP, DEKR48 Geng et al. (2021b) by 0.026 AP, and Pose*Plus*Seg Ahmad et al. (2022) by 0.005.

Table 2 presents the performance of CENTERFOCUS using the COCO keypoint *test* set, outperforms SOTA bottom-up techniques including HGG Jin et al. (2020), Point-Set Anchors Wei et al.

Models	Backbone	AP	AP$^{.50}$	AP$^{.75}$	AP^{M}	AP^{L}	AR
Top-down:							
Mask-RCNN	ResNet-50-FPN	0.631	0.873	0.687	0.578	0.714	-
G-RMI COCO-only	ResNet-101	0.649	0.855	0.713	0.623	0.700	0.697
Integral Pose Regression	ResNet-101	0.678	0.882	0.748	0.639	0.740	-
G-RMI + extra data	ResNet-101	0.685	0.871	0.755	0.658	0.733	0.733
CPN	ResNet-50	0.721	0.914	0.800	0.687	0.772	0.785
RMPE	PyraNet	0.723	0.892	0.791	0.680	0.786	-
CFN	-	0.726	0.861	0.697	0.783	0.641	-
CPN (ensemble)	ResNet-Inception	0.730	0.917	0.809	0.695	0.781	0.790
HRNet-W48	HRNet-W48	0.755	0.925	0.833	0.719	0.815	0.790
Bottom-up:							
OpenPose ∗	-	0.618	0.849	0.675	0.571	0.682	0.665
Directpose †	ResNet-101	0.648	0.878	0.711	0.604	0.715	-
Ass. Emb. † ∗	Hourglass	0.655	0.868	0.723	0.606	0.726	0.702
PifPaf	ResNet-152	0.667	-	-	0.624	0.729	0.722
SPM	Hourglass	0.669	0.885	0.729	0.626	0.731	-
PersonLab †	ResNet-152	0.687	0.890	0.754	0.641	0.755	0.754
MultiPoseNet	ResNet-101	0.696	0.863	0.766	0.650	0.763	0.735
HGG †	Hourglass	0.676	0.851	0.737	0.627	0.746	0.713
Point-Set Anchors †	HRNet-W48	0.687	0.899	0.763	0.648	0.753	-
HigherHRNet †	HRNet-W48	0.705	0.893	0.772	0.666	0.758	0.749
SIMPLE-W32 †	HRNet-W32	0.711	0.902	0.794	0.691	0.791	-
DEKR †	HRNet-W48	0.723	0.883	0.786	0.686	0.786	0.777
Pose*Plus*Seg	ResNet-152	0.728	0.884	0.787	0.678	0.794	0.798
CenterFocus (ours)	ResNet-101	0.722	0.870	0.772	0.663	0.791	0.788
CenterFocus (ours)	ResNet-152	0.731	0.889	0.789	0.681	0.795	0.804

Table 2: Performance comparison using the **COCO** keypoint *test* set. ∗ indicates refinement. † indicates multi-scale test.

(2020), HigherHRNet Cheng et al. (2020), SIMPLE Zhang et al. (2021), DEKR Geng et al. (2021a), and Pose*Plus*Seg Ahmad et al. (2022). Specifically, CenterFocus yields a mAP of 0.731 using ResNet-152 as a backbone feature extractor.

Table 3 demonstrates the results of our proposed model CenterFocus against SOTA models using the OCHuman challenging dataset. We compared keypoint prediction accuracy with SOTA approaches: HGG Jin et al. (2020) and MIPNet Khirodkar et al. (2021) using the OCHuman *val* and *test* sets. CenterFocus improves 0.07 AP over HGG Jin et al. (2020) and 0.005 AP over MIPNet Khirodkar et al. (2021) in the *test* set.

Models	Backbone	Val mAP	Test mAP
HGG	Hourglass	0.356	0.348
HGG †	Hourglass	0.418	0.360
MIPNet	ResNet101	0.420	0.425
CenterFocus (ours)	ResNet101	0.434	0.429
CenterFocus (ours)	ResNet152	0.439	0.430

Table 3: Performance comparison using **OCHuman** keypoint *val* and *test* datasets. † indicates multi-scale testing.

Models	Backbone	AP	AP$^{.50}$	AP$^{.75}$	AP^{M}	AP^{L}	AR
PersonLab ×	ResNet101	0.382	0.661	0.397	0.476	0.592	0.162
PersonLab ×	ResNet152	0.387	0.667	0.406	0.483	0.595	0.163
PersonLab †	ResNet101	0.414	0.684	0.447	0.492	0.621	0.170
PersonLab †	ResNet152	0.418	0.688	0.455	0.497	0.621	0.170
Pose2Seg	ResNet50-fpn	0.555	-	-	0.498	0.670	-
Pose*Plus*Seg	ResNet-152	0.563	0.701	0.557	0.509	0.683	0.701
CenterFocus (ours)	ResNet101	0.559	0.721	0.559	0.521	0.691	0.699
CenterFocus (ours)	ResNet152	0.570	0.724	0.565	0.547	0.698	0.706

Table 4: Performance comparison using the **COCO** Segmentation *val* set. × indicates single-scale testing. † indicates multi-scale testing.

Segmentation Results. Table 4 and Table 5 present the results of COCO Segmentation *val* and *test* sets. CenterFocus achieved a mAP of 0.570 on the *val* set, and improved the AP by 0.152 compared with PersonLab Papandreou et al. (2018), 0.015 AP compared with Pose2Seg Zhang et al. (2019), and 0.007 compared with Pose*Plus*Seg Ahmad et al. (2022). Moreover, on the test set, CenterFocus achieved a mAP of 0.456 and improved the AP by 0.085 over Mask-RCNN He et al. (2017), 0.039 over PersonLabPapandreou et al. (2018), and 0.011 over Pose*Plus*Seg Ahmad et al. (2022). Table 6 shows segmentation performance compared with Pose2Seg Zhang et al. (2019) using the OCHuman *val* and *test* sets.

Models	Backbone	AP	AP$^{.50}$	AP$^{.75}$	AP^{M}	AP^{L}	AR
Mask-RCNN	ResNeXt-101	0.371	0.600	0.394	0.399	0.535	-
PersonLab ×	ResNet101	0.377	0.659	0.394	0.480	0.595	0.162
PersonLab ×	ResNet152	0.385	0.668	0.404	0.488	0.602	0.164
PersonLab †	ResNet101	0.411	0.686	0.445	0.496	0.626	0.169
PersonLab †	ResNet152	0.417	0.691	0.453	0.502	0.630	0.171
Pose*Plus*Seg	ResNet152	0.445	0.794	0.471	0.524	0.651	0.677
CenterFocus (ours)	ResNet101	0.439	0.804	0.478	0.535	0.674	0.677
CenterFocus (ours)	ResNet152	0.456	0.818	0.487	0.546	0.678	0.682

Table 5: Performance comparison using the **COCO** Segmentation *test* set. × indicates single-scale testing. † indicates multi-scale testing.

Models	Backbone	Val mAP	Test mAP
Pose2Seg	ResNet50-fpn	0.544	0.552
CenterFocus (ours)	ResNet101	0.555	0.550
CenterFocus (ours)	ResNet152	0.559	0.556

Table 6: Performance comparison using the **OCHuman** segmentation val and test datasets.

Computation Cost and Inference Time. We calculate the computational cost in GFLOPs and the number of parameters for an approximate image size of 401 × 401 resolution. Table 7 shows that CenterFocus with ResNet-50 has the lowest FLOPs and highest mAP compared with Hourglass Newell et al. (2016) and CPN Chen et al. (2018). CenterFocus with ResNet-101 and ResNet-152 also incurs lower FLOPs and number of parameters compared with the representative top competitors, including SimpleBaseline Xiao et al. (2018), HRNet Sun et al. (2019), DEKR Geng et al. (2021a), PersonLab Papandreou et al. (2018), and Pose*Plus*Seg Ahmad et al. (2022).

Table 10 in the Appendix presents the inference time and runtime measurements of CenterFocus on a single GPU (Titan RTX).

Models	Backbone	Input Size	GFLOPs	# Parameters	mAP
Hourglass	8-stage	256 × 192	14.3	25.1M	0.669
CPN	ResNet-50	256 × 192	6.20	27.0M	0.686
CPN*	ResNet-50	384 × 288	6.20	27.0M	0.694
SimpleBaseline	ResNet-152	256 × 192	15.7	68.6M	0.720
HrHRNet	HRNet-W48	640 × 640	154.3	63.8M	0.723
DEKR48	HRNet-W48	640 × 640	141.5	65.7M	0.710
PersonLab	ResNet-101	1401 × 1401	405.5	68.7M	0.665
Pose*Plus*Seg	ResNet-152	256 × 192	11.34	60.1M	0.744
CenterFocus	ResNet-50	≈401	4.04	25.5M	0.697
CenterFocus	ResNet-101	≈401	7.69	44.5M	0.736
CenterFocus	ResNet-152	≈401	11.34	60.1M	0.751

Table 7: Comparison of GFLOPs and # parameters on the val set. CPN* is with online hard keypoints mining.

5 Ablation Experiments

5.1 SKFM and KeyCentroid

We first compare the SKFM with keypoint detection algorithms relying on keypoint feature maps to qualitatively analyze SKFM. Table 8 presents the performance of SKFM and KeyCentroid (k_c) with the SOTA bottom-up approaches including CMU-Pose Cao et al. (2017), MultiPoseNet Kocabas et al. (2018), PersonLab Papandreou et al. (2018), HGG Jin et al. (2020), SimpleBaseline Xiao et al. (2018), and Pose*Plus*Seg Ahmad et al. (2022). The SKFM generated with the help of a keypoint disk outperforms the SOTA methods, yielding a mAP of 0.725 using ResNet152. In addition, we define KeyCentroids and aggregated them with the SKFM to find the optimal 2D keypoint value. Thus, CenterFocus further improved the keypoint accuracy to 0.024 points and secured a 0.749 mAP resulting in a significant improvement over Pose*Plus*Seg Ahmad et al. (2022).

5.2 Keypoint Confidence Scoring

We assess the BHM on the keypoint confidence score prediction. Figure 6 shows the 17 keypoint detection confidence scores generated by the keypoint disks at radius R = 8, 16, and 32. The keypoint detection confidence score for a large radius (R = 32) is high because it provides optimal space for the classifier to reach the local minimum.

Models	AP	AP$^{.50}$	AP$^{.75}$	AP^{M}	AP^{L}
CMU-Pose	0.610	0.849	0.675	0.563	0.693
MultiPoseNet	0.643	0.882	0.750	0.596	0.739
PersonLab	0.665	0.862	0.719	0.623	0.732
HGG	0.683	0.867	0.758	-	-
SimpleBaseline	0.720	0.893	0.798	0.687	0.789
Pose*Plus*Seg	0.744	0.894	0.748	0.675	0.811
CenterFocus (ours):					
ResNet101 (SKFM)	0.717	0.782	0.726	0.611	0.776
ResNet152 (SKFM)	0.725	0.897	0.816	0.703	0.791
ResNet101 (SKFM + k$_c$)	0.735	0.898	0.819	0.715	0.797
ResNet152 (SKFM + k$_c$)	0.749	0.899	0.824	0.718	0.816

Table 8: Keypoint detection comparison between CenterFocus SKFM and SKFM + k$_c$ variant with existing keypoint feature maps approaches.

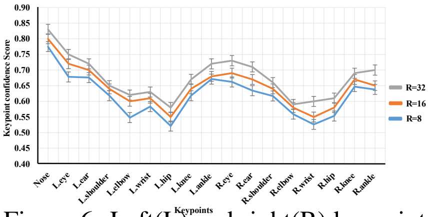


Figure 6: Left(L) and right(R) keypoint confidence score with disk radius R.

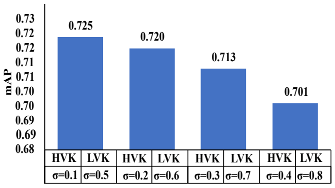


Figure 7: Keypoint detection results with varied σ values.

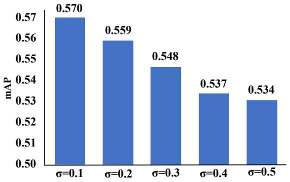


Figure 8: Segmentation results with varied σ values.

5.3 Analyzing MaskCentroid

We analyze the MaskCentroid that plays an important role in human instance segmentation by defining centroids as a center of attraction for the pixels in the embedding space. We compared CenterFocus's MaskCentroid with human segmentation models. Table 9 shows the MaskCentroid accuracy trade-off compared with PersonLab Papandreou et al. (2018), Pose2Seg Zhang et al. (2019), and Pose*Plus*Seg Ahmad et al. (2022).

Models	AP	AP$^{.50}$	AP$^{.75}$	AP^{M}	AP^{L}
PersonLab	0.418	0.688	0.455	0.497	0.621
Pose2Seg	0.555	-	-	0.498	0.670
Pose*Plus*Seg	0.563	0.701	0.557	0.509	0.683
CenterFocus:					
ResNet101	0.550	0.705	0.548	0.518	0.685
ResNet152	0.558	0.711	0.558	0.528	0.691
ResNet101 (MaskCentroid)	0.559	0.721	0.559	0.531	0.690
ResNet152 (MaskCentroid)	0.570	0.724	0.565	0.547	0.698

Table 9: MaskCentroid performance using the COCO Segmentation.

5.4 Impact of Point-wise and Instance-wise Gaussian Optimization

We generated SKFM using $0<\sigma<1$ for different keypoints, i.e., *point-wise Gaussian optimization*. Figure 7 summarizes the mAP for different σ with high variation keypoints (HVK) and low variation keypoints (LVK).

Finally, we examine the impact of *instance-wise Gaussian optimization* on instance segmentation task. We tested the sensitivity of σ ranging from 0.1 to 0.5 on human instance segmentation. Figure 8 shows the results with different σ values, where low σ performs better in crowded cases.

6 Conclusion

We propose CenterFocus, a bottom-up reliable approach, to tackle the task of human pose estimation and instance segmentation. CenterFocus introduced a Strong Keypoint Feature Map and KeyCentroid to find the optimal 2D keypoint position. In addition, a MaskCentroid was used that defines the keypoint as a centroid in the embedding space to associate the pixels to the right instance for instance-level segmentation. The effectiveness of CenterFocus was tested using the COCO and OCHuman challenging datasets and showed incredible performance.

REFERENCES

Niaz Ahmad, Jawad Khan, Jeremy Yuhyun Kim, and Youngmoon Lee. Joint human pose estimation and instance segmentation with poseplusseg. In *AAAI*, 2022. 2, 3, 6, 7, 8, 9, 13

Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 3

Yuanhao Cai, Zhicheng Wang, Zhengxiong Luo, Binyi Yin, Angang Du, Haoqian Wang, Xiangyu Zhang, Xinyu Zhou, Erjin Zhou, and Jian Sun. Learning delicate local representations for multi-person pose estimation. In *European Conference on Computer Vision*, pp. 455–472. Springer, 2020. 3

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 3, 8

Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018. 1, 2, 3, 6, 8

Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5386–5395, 2020. 3, 7

Moo K Chung. Gaussian kernel smoothing. *arXiv preprint arXiv:2007.09539*, 2020. 4, 6

Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *ECCV*, 2016. 3

Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *ICCV*, 2017. 1, 2, 3

Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14676–14686, 2021a. 7, 8

Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14676–14686, 2021b. 6

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6, 13

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2, 3, 7, 13

Shaoli Huang, Mingming Gong, and Dacheng Tao. A coarse-fine network for keypoint localization. In *ICCV*, 2017. 1, 2

Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016. 3

Sheng Jin, Wentao Liu, Enze Xie, Wenhai Wang, Chen Qian, Wanli Ouyang, and Ping Luo. Differentiable hierarchical graph grouping for multi-person pose estimation. In *European Conference on Computer Vision*, pp. 718–734. Springer, 2020. 3, 6, 7, 8

Rawal Khirodkar, Visesh Chari, Amit Agrawal, and Ambrish Tyagi. Multi-hypothesis pose networks: Rethinking top-down pose estimation. *arXiv preprint arXiv:2101.11223*, 2021. 7

Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *ECCV*, 2018. 3, 8, 13

Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11977–11986, 2019. 3, 6

Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011. 6

Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019. 1, 2

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 2, 6, 13, 14, 15

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, pp. 8759–8768, 2018. 3

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 3, 6, 8

Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NeurIPS*, 2017. 3

George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *ECCV*, 2018. 1, 2, 3, 7, 8, 9

Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016. 3

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 3

Kai Su, Dongdong Yu, Zhenqi Xu, Xin Geng, and Changhu Wang. Multi-person pose estimation with enhanced channel-wise and spatial information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5674–5682, 2019. 3

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5693–5703, 2019. 8

Jian Wang, Xiang Long, Yuan Gao, Errui Ding, and Shilei Wen. Graph-pcnn: Two stage human pose estimation with graph pose refinement. In *European Conference on Computer Vision*, pp. 492–508. Springer, 2020. 3

Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. In *European Conference on Computer Vision*, pp. 527–544. Springer, 2020. 6

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 3

Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 466–481, 2018. 3, 6, 8

Jiabin Zhang, Zheng Zhu, Jiwen Lu, Junjie Huang, Guan Huang, and Jie Zhou. Simple: Single-network with mimicking and point learning for bottom-up human pose estimation. *arXiv preprint arXiv:2104.02486*, 2021. 7

Song-Hai Zhang, Ruilong Li, Xin Dong, Paul Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shi-Min Hu. Pose2seg: Detection free human instance segmentation. In *CVPR*, 2019. 1, 2, 3, 6, 7, 9, 13

Chunluan Zhou and Junsong Yuan. Multi-label learning of part detectors for heavily occluded pedestrian detection. In *ICCV*, pp. 3486–3495, 2017. 5

Chunluan Zhou and Junsong Yuan. Bi-box regression for pedestrian detection and occlusion estimation. In *ECCV*, pp. 135–151, 2018. 5

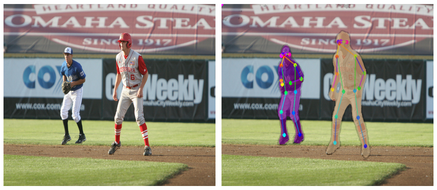


Figure 9: Pose estimation and instance segmentation.

A Appendix

A.1 Pose and Instance Segmentation

We introduce a new PoseSeg algorithm to present the human pose estimation along with the instance segmentation, as visualized in Figure 9. The PoseSeg module uses the features generated from both head networks to get the final pose and instance segmentation as shown in Figure 2. At first, the keypoints and their 2D coordinates are cached in a priority queue. These keypoints are then used to make a body skeleton by gradually connecting keypoints with common vertices. If two keypoints are overlapped, entangled, or more than one keypoint (x and y coordinates) are identified for a single keypoint position p_i, non-maximum suppression is leveraged to select the final keypoint coordinates. Next, a new instance k' starts with the j^{th} keypoint detected at image position x_i and is considered a new point.

CenterFocus also performs instance-level segmentation for all detected human instances. It identifies pixel positions p_i belonging to an instance, i.e., those pixels with the highest probability that lie in the embedding space. These pixels are then assigned to the relevant instance if the pixel embedding is close to the centroid of the instance. Specifically, if the probability P of a pixel p_i, $P(p_i) \geq 0.5$, then the pixel at position x_i is assigned to the relevant human instance. We assume that pixels with a probability threshold ≥ 0.5 are close to the centroid of the instance and are considered as part of a particular instance. Otherwise, the pixels belong to another instance or background.

A.2 Inference Time

The measure system inference time is compared with SOTA multi-task models, including Mask-RCNN He et al. (2017), Pose2seg Zhang et al. (2019), MultiPoseNet Kocabas et al. (2018), and Pose*Plus*Seg Ahmad et al. (2022). Table 10 presents the inference time and runtime measurements of CenterFocus on a single GPU (Titan RTX), demonstrating its efficiency in processing for human pose, instance segmentation, and joint human pose and segmentation.

Models	Backbone	Task	Inference Time	GPU
Mask R-CNN	ResNet-101	Boxes, segmentation & pose	200ms (5fps)	Tesla M40
Pose2Seg	ResNet50-fpn	Instance segmentation	50ms (20fps)	GTX Titan X
MultiPoseNet	ResNet-101	pose estimation	43ms (23fps)	GTX 1080Ti
Pose*Plus*Seg	ResNet-152	Pose & segmentation	34ms (28fps)	Titan RTX
CenterFocus	ResNet-152	Pose estimation	28ms (35fps)	Titan RTX
CenterFocus	ResNet-152	Instance segmentation	30ms (32fps)	Titan RTX
CenterFocus	ResNet-152	Pose & segmentation	34ms (29fps)	Titan RTX

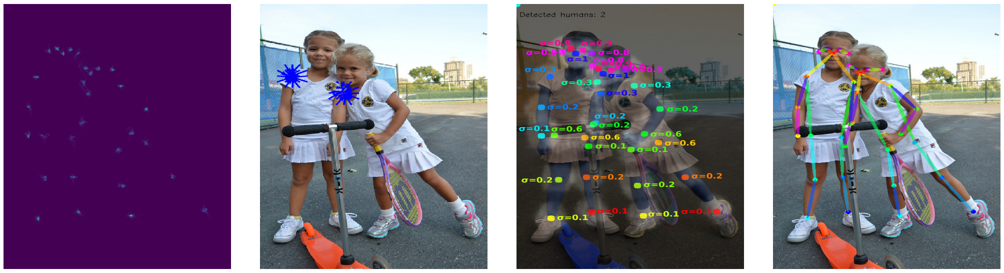
Table 10: Comparison of the runtime performance.

A.3 Real-world visualization

We investigate the significance of each component of CenterFocus and illustrate them visually. All the results are generated from the same resolution defined in the COCO Lin et al. (2014) dataset using the ResNet 152 He et al. (2016). Section A.4 visualizes the impact of Strong Keypoint Feature Map (SFHM), *Point-wise Gaussian Optimization* (PGO), KeyCentroid, and Body Heat Map (BHM) on human pose estimation. Section A.5 illustrates the impacts of MaskCentroid and *Instance-wise Gaussian Optimization* (IGO) on instance segmentation. Finally, Section A.6 displays the examples of human pose along with instance segmentation.

A.4 Pose Estimation

Figure 10 illustrates examples of pose estimation generated from different components of the system. Figure 11 shows examples of pose estimation results. Figure 12 shows PGO along with body heat map from COCO Lin et al. (2014) *val* dataset.



(a) SKFM (b) KeyCentroid (c) PGO+BHM (d) Pose

Figure 10: (a) shows SKFM generated from Pose Head network, (b) shows KeyCentroid to find optimal 2D keypoint, (c) shows the PGO by defining the optimal σ values for high and low variant keypoints, and (d) defines the final pose estimation.

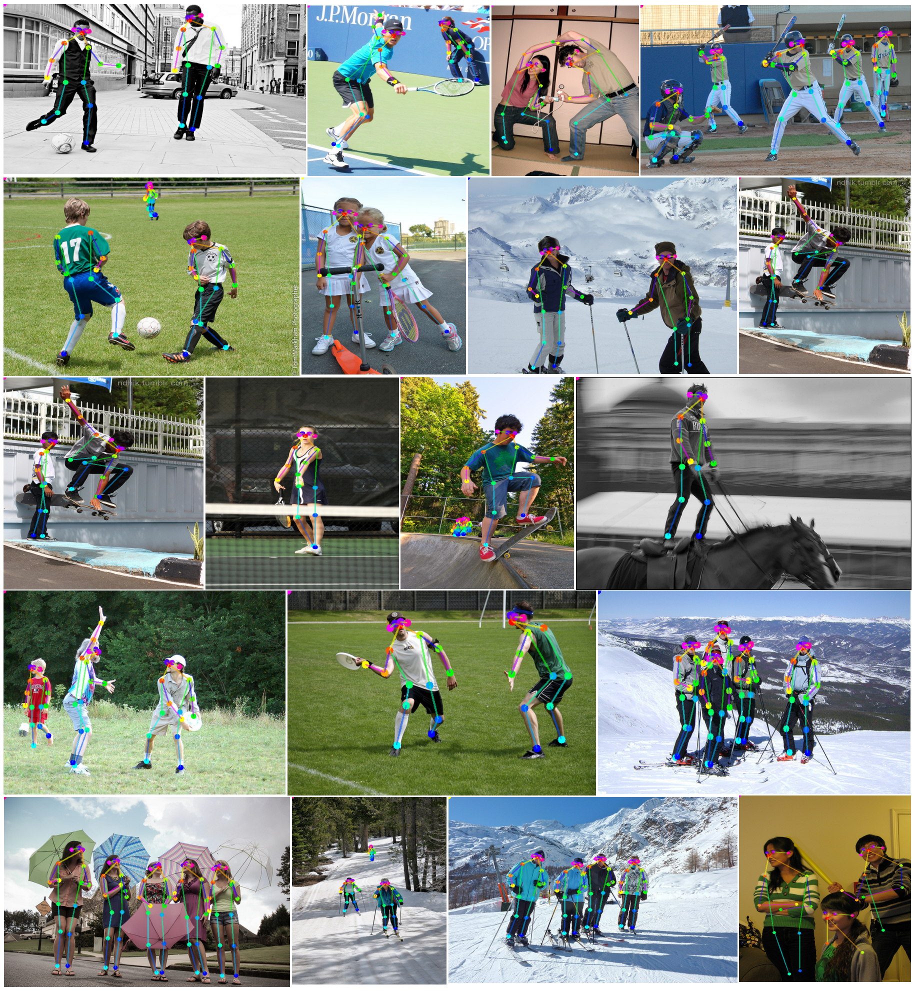


Figure 11: Pose estimation examples illustrating keypoints.

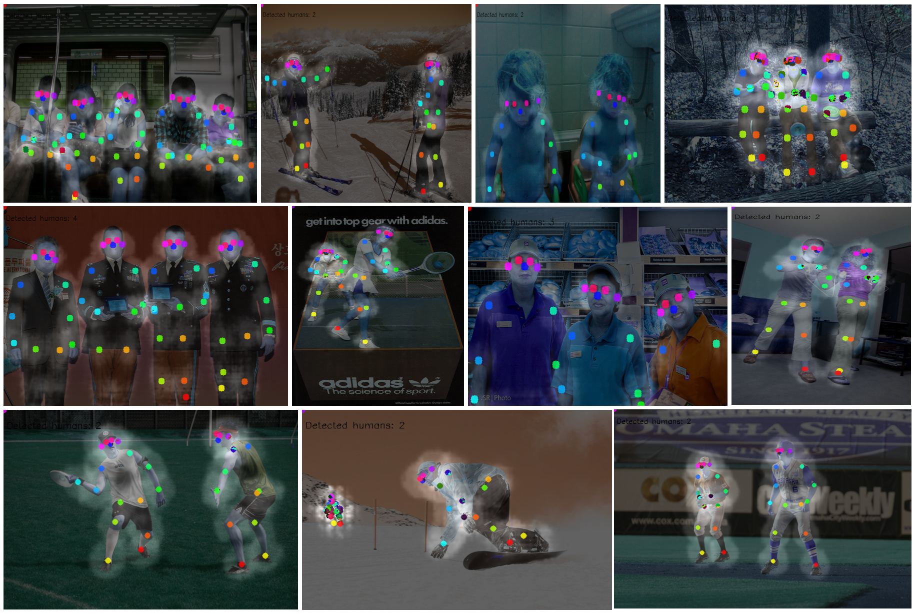
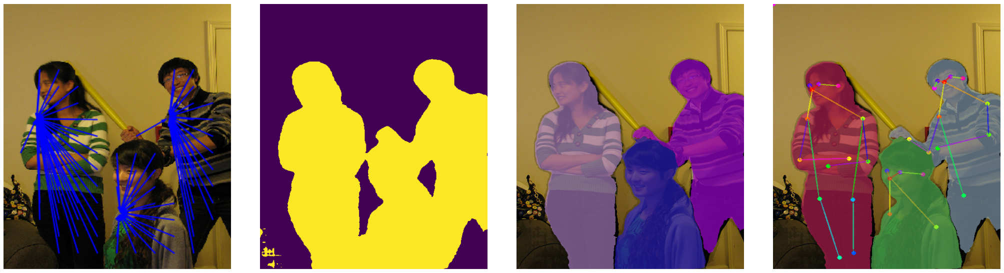


Figure 12: Point-wise Gaussian optimization along with body heat map.

A.5 Instance Segmentation

Figure 13 illustrates examples of instance segmentation generated at each stage of the system. Figure 14 shows examples of the instance segmentation from COCO Lin et al. (2014) *val* dataset.



(a) MaskCentroid (b) SFM (c) IGO + Instance Segmentation (d) Pose & Instance Segmentation

Figure 13: (a) shows MaskCentroid to assign pixels to a particular human instance, (b) shows Semantic Feature Map (SFM) generated from the Segmentation Head network, (c) defines the instance segmentation after applying the IGO, and (d) combines the pose estimation and instance segmentation.

A.6 Pose and Instance Segmentation

Figure 15 illustrates examples of the pose estimation and instance segmentation from the COCO Lin et al. (2014) *val* dataset.

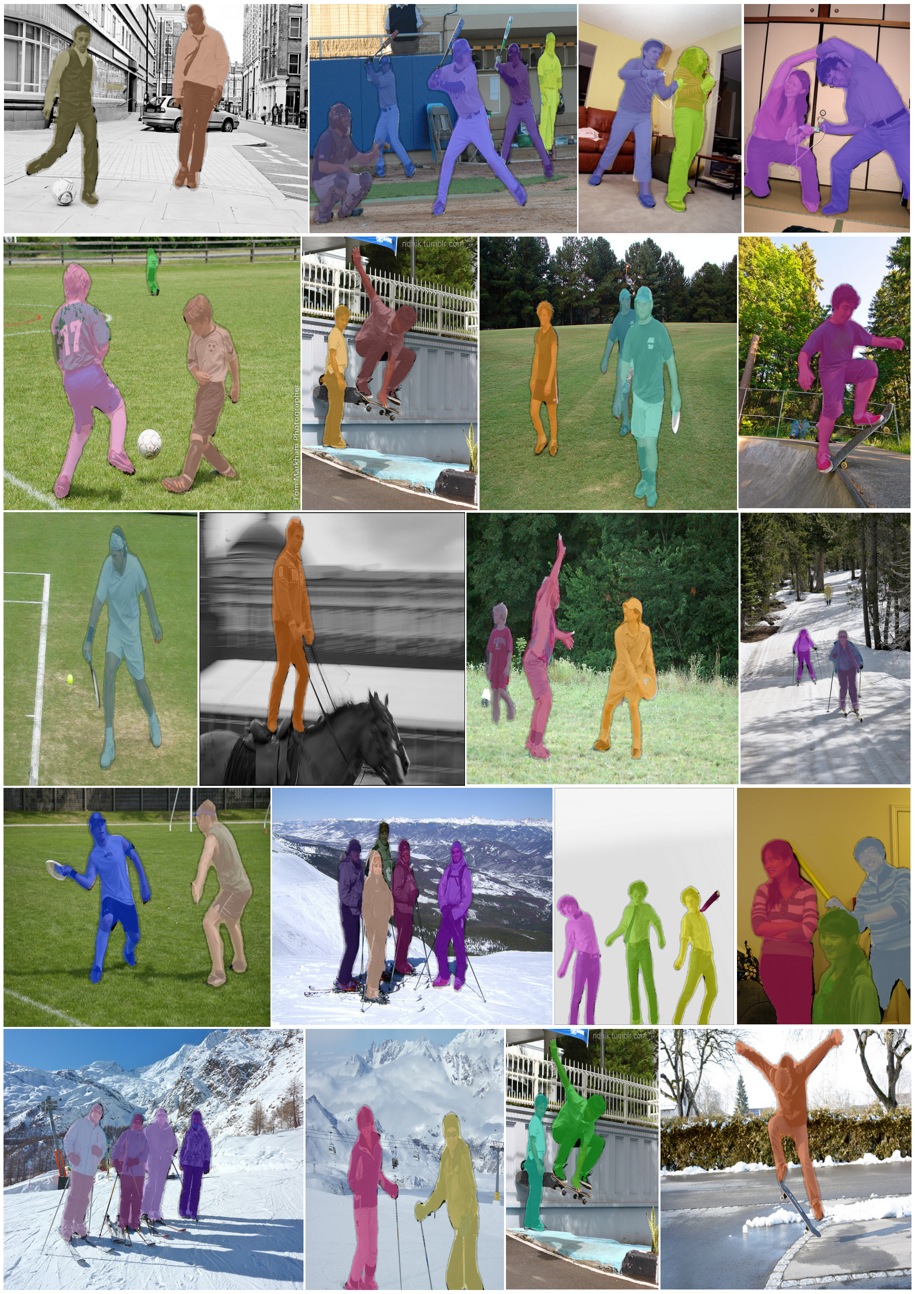


Figure 14: Instance segmentation.

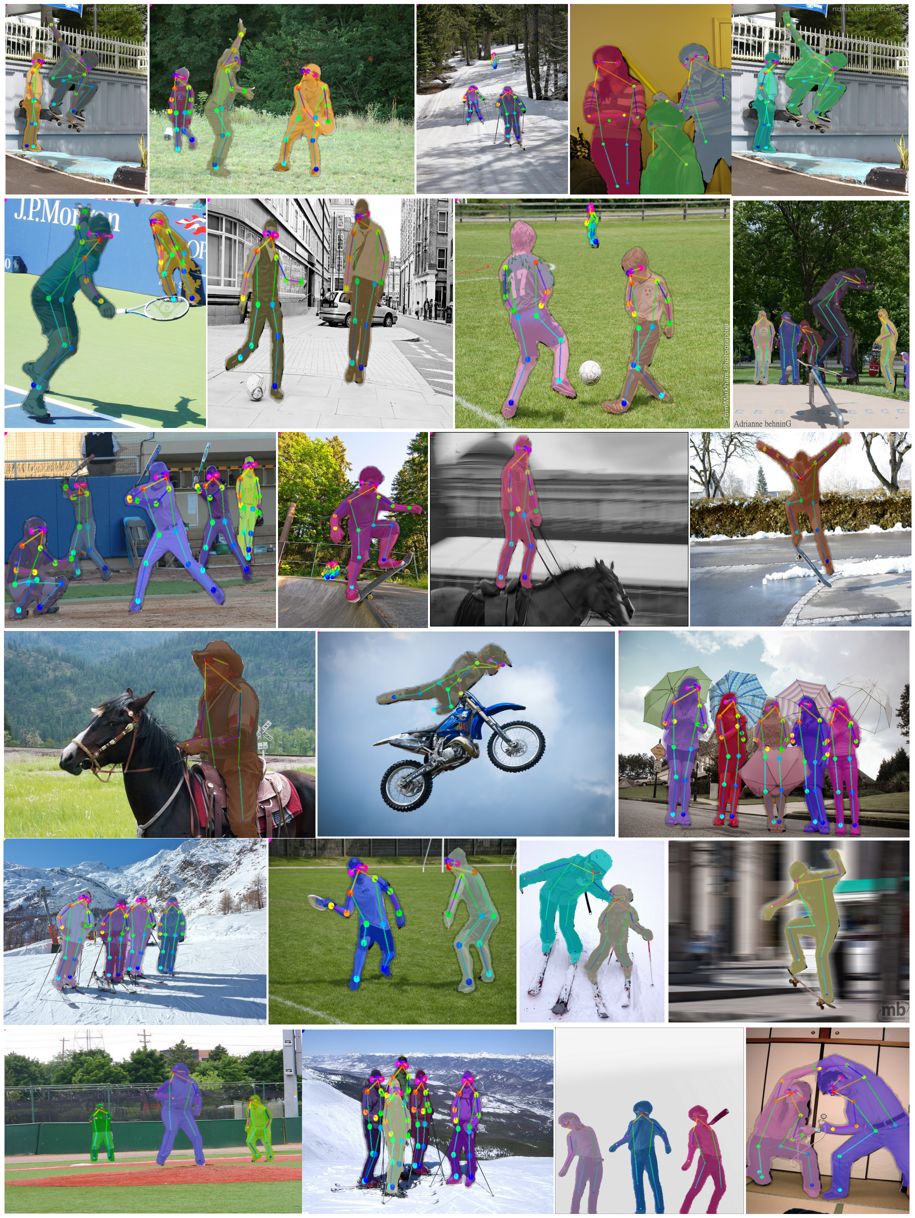


Figure 15: Pose and instance segmentation.